

Amendments to the Claims

This listing of claims will replace all prior versions, and listings of claims in the application.

1. (Currently Amended) A method for in-place, lightweight Ack packet promotion, comprising:

receiving a new Ack packet;

searching through a transmit queue for an old Ack packet that corresponds to the new Ack packet; ~~and~~

replacing the data in a number field, a checksum field, a window size field, and/or a timestamp options field of the old Ack packet with data in a number field, a checksum field, a window size field, and/or a timestamp options field of the new Ack packet; and

queuing the new Ack packet in response to detecting that the old Ack packet has been transmitted before said replacing step is completed.

2. (original) The method of claim 1, further comprising:

discarding the new Ack packet.

3. (original) The method of claim 1, further comprising:

queuing the new Ack packet for a piggyback request.

4. (Canceled)

5. (original) The method of claim 1, further comprising:

determining whether a Transmission Control Protocol (TCP) stream associated with the old Ack packet and a TCP stream associated with the new Ack packet are from the same stream.

6. (original) The method of claim 5, wherein said determining step comprises identifying the old Ack packet TCP stream and the new Ack packet TCP stream from a tuple.

7. (original) The method of claim 6, wherein the tuple includes a source IP address, a destination IP address, a source port number, and a destination port number.

8. (original) The method of claim 5, further comprising:
utilizing a hash algorithm to execute said determining step, wherein the hash algorithm includes the checksum fields of the old Ack packet and the new Ack packet.

9. (original) The method of claim 8, further comprising:
utilizing a source port number in the hash algorithm to execute said determining step, wherein the source port number identifies a source of a corresponding TCP stream.

10. (original) The method of claim 1, further comprising:
queuing the new Ack packet for transmission if a corresponding old Ack packet is not found.

11. (original) A system for in-place Ack packet promotion, comprising:

a cable modem in communications with a cable network and configured to receive one or more Ack packets for delivery over said cable network; and

a transmit queue coupled to said cable modem, wherein said transmit queue is searchable to compare a new Ack packet with an older Ack packet that corresponds to said new Ack packet, wherein said cable modem operates to replace the data in a number field, a checksum field, a window size field, and/or a timestamp options field of said older Ack packet with data in a number field, a checksum field, a window size field, and/or a timestamp options field of said new Ack packet, wherein said cable modem operates to queue the new Ack packet in response to detecting that the old Ack packet has been transmitted before said replacing step is completed.

12. (original) The system of claim 11, wherein a hash algorithm, when executed, determines whether a Transmission Control Protocol (TCP) stream associated with said older Ack packet and a TCP stream associated with said new Ack packet are from the same stream.

13. (Currently Amended) A computer program product comprising a computer useable medium having a computer readable program code means embedded in said medium for causing a computer to manage a system for in-place Ack packet promotion, comprising:

first computer readable program code means for receiving a new Ack packet;

second computer readable program code means for searching through a transmit queue for an old Ack packet that corresponds to said new Ack packet; ~~and~~

third computer readable program code means for replacing the data in a number field, a checksum field, a window size field, and/or a timestamp options field of said old Ack packet with data in a number field, a checksum field, a window size field, and/or a timestamp options field of said new Ack packet; and

fourth computer readable program code means for queuing the new Ack packet in response to detecting that the old Ack packet has been transmitted before said replacing step is completed.

14. (Currently Amended) The computer program product according to claim 13, further comprising:

~~fourth~~ fifth computer readable program code means for discarding said new Ack packet.

15. (Currently Amended) The computer program product according to claim 13, further comprising:

~~fourth~~ fifth computer readable program code means for determining whether a TCP stream associated with said old Ack packet and a TCP stream associated with said new Ack packet are from the same stream.

16. (Original) The computer program product according to claim 15, wherein a tuple is utilized to identify said old Ack packet TCP stream and said new Ack packet TCP stream.

17. (original) The computer program product according to claim 16, wherein said tuple includes a source IP address, a destination IP address, a source port number, and a destination port number.

18. (original) The computer program product according to claim 15, wherein the checksum fields of the said Ack packet and said new Ack packet are used as a hash algorithm.

19. (Currently Amended) The computer program product according to claim 13, wherein said first computer readable program code means, said second computer readable program code means, ~~and~~ said third computer readable program code means, and said fourth computer readable program code means operate within a cable modem.